

RTI Queuing Service

Release Notes

Version 7.3.0



Trademarks

RTI, Real-Time Innovations, Connex, Connex Drive, NDDS, the RTI logo, 1RTI and the phrase, “Your Systems. Working as one.” are registered trademarks, trademarks or service marks of Real-Time Innovations, Inc. All other trademarks belong to their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished solely under and subject to RTI's standard terms and conditions available at <https://www.rti.com/terms> and in accordance with your License Acknowledgement Certificate (LAC) and Maintenance and Support Certificate (MSC), except to the extent otherwise accepted in writing by a corporate officer of RTI.

Third-Party Software

RTI software may contain independent, third-party software or code that are subject to third-party license terms and conditions, including open source license terms and conditions. Copies of applicable third-party licenses and notices are located at community.rti.com/documentation. IT IS YOUR RESPONSIBILITY TO ENSURE THAT YOUR USE OF THIRD-PARTY SOFTWARE COMPLIES WITH THE CORRESPONDING THIRD-PARTY LICENSE TERMS AND CONDITIONS.

Notices

Deprecations and Removals

Any deprecations or removals noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220 and/or any other agreements by and between RTI and customer regarding maintenance and support of RTI's software.

Deprecated means that the item is still supported in the release, but will be removed in a future release. *Removed* means that the item is discontinued or no longer supported. By specifying that an item is deprecated in a release, RTI hereby provides customer notice that RTI reserves the right after one year from the date of such release and, with or without further notice, to immediately terminate maintenance (including without limitation, providing updates and upgrades) for the item, and no longer support the item, in a future release.

Technical Support

Real-Time Innovations, Inc.

232 E. Java Drive

Sunnyvale, CA 94089

Phone: (408) 990-7444

Email: support@rti.com

Website: <https://support.rti.com/>

Contents

| | |
|---|-----------|
| 1 Supported Platforms | 1 |
| 2 Compatibility | 2 |
| 3 What's New in 7.3.0 LTS | |
| 3.1 Experimental Product Status (Reminder) | 3 |
| 3.2 Third-Party Software Upgrades | 3 |
| 3.3 New Platform for Unreal Engine 5.2 | 4 |
| 4 What's Fixed in 7.3.0 LTS | |
| 4.1 [Critical] Potential Arbitrary SQL Query Execution while Parsing Malicious Remote Commands or Configuration Files | 5 |
| 4.2 [Critical] Configuring SharedReaderQueue's DataWriter QoS to use compression and XCDR2 encapsulation caused samples to not be received by Consumers | 6 |
| 4.3 [Critical] Queuing Service sent samples with wrong data representation to Consumers | 6 |
| 4.4 [Major] Lifespan per Producer not applied when Producer sent data on behalf of another Producer's DataWriter | 7 |
| 4.5 [Minor] Warnings when using Queuing Service with replication | 7 |
| 4.6 [Minor] Fourth digit of product version not logged by Queuing Service at startup | 8 |
| 5 Known Issues | |
| 5.0.1 Some tags in the XML configuration must be grouped in a strict order | 9 |
| 6 Current Limitations | 10 |
| 7 Available Documentation | 11 |

1 Supported Platforms

See [Supported Platforms, in the RTI Connex Core Libraries Release Notes](#).

2 Compatibility

Queuing Service is built on top of, and intended for use with, *RTI Connex*® with the same version number.

For backward compatibility information, if any, between 7.3.0 and previous releases, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

3 What's New in 7.3.0 LTS

3.1 Experimental Product Status (Reminder)

Queuing Service has been an experimental product since [release 6.1.2](#). Whereas in [previous 7.x feature releases](#), *Queuing Service* was not included with the release, it is included again in release 7.3.0 and is still experimental.

As with all RTI experimental products, *Queuing Service* in release 6.1.2 and above should not be used in production applications. See *Experimental Features* in the [RTI Connex7 Core Libraries Release Notes](#) for information on RTI experimental products and features.

If you already have *Queuing Service* from another release (6.1.1 or earlier) in which it is fully supported, you can continue using it in that release. If you use an earlier version of *Queuing Service* with a *Connex7* release, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>); look for the "Wire Protocol" sections to find wire protocol compatibility issues, if any, between the older release and the *Connex7* release. Address these issues before using an older version of *Queuing Service* with *Connex7*.

3.2 Third-Party Software Upgrades

The following third-party software used by *Queuing Service* has been upgraded since release 6.1.2:

| Third-Party Software | Previous Version (in <i>Connex</i> 6.1.2) | Current Version |
|----------------------|---|-----------------|
| libxslt | 1.1.34 | 1.1.38 |
| libxml2 | 2.9.12 | 2.11.4 |
| SQLite | 3.37.2 | 3.39.0 |

3.3 New Platform for Unreal Engine 5.2

This release adds the x64Linux5Unreal5.2clang15 architecture, with support for *Queuing Service*. This architecture targets Unreal Engine® 5.2 on Ubuntu 22.04 LTS for x64 CPUs. See the [RTI Connex Core Libraries Platform Notes](#) for details on using this architecture.

For a list of all new platforms in this release, see the [RTI Connex What's New](#).

4 What's Fixed in 7.3.0 LTS

The following issues have been fixed since 6.1.2.

[Critical]: System-stopping issue, such as a crash or data loss.

[Major]: Significant issue with no easy workaround.

[Minor]: Issue that usually has a workaround.

[Trivial]: Small issue, such as a typo in a log.

4.1 **[Critical]** Potential Arbitrary SQL Query Execution while Parsing Malicious Remote Commands or Configuration Files

There was the potential for arbitrary SQL query execution while parsing malicious remote administration commands or loading a malicious configuration file. This vulnerability is now fixed.

User Impact without Security

A SQL Injection vulnerability in *Queuing Service* could have resulted in the following:

- Arbitrary SQL query execution.
- Remotely exploitable.
- Potential impact on integrity and confidentiality of *Queuing Service*.
- CVSS Base Score: 9.1 CRITICAL
- CVSS v3.1 Vector: [AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N](#)

User Impact with Security

When enabling RTPS protection, the impact of the SQL Injection vulnerability in Queuing Service was reduced, resulting in the following:

- Arbitrary SQL query execution.
- Exploitable from the same host *Queuing Service* is running.
- Potential impact on integrity and confidentiality of *Queuing Service*.
- CVSS Base Score: 7.1 HIGH
- CVSS v3.1 Vector: [AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N](#)

[RTI Issue ID QUEUEING-756]

4.2 [Critical] Configuring SharedReaderQueue's DataWriter QoS to use compression and XCDR2 encapsulation caused samples to not be received by Consumers

Using XCDR2 encapsulation and compression to configure the *DataWriter* QoS of a *SharedReaderQueue* may have led to samples not being received by the Consumers' *DataReaders*.

When this problem occurred, you may have seen the following error messages on the Consumer's *DataReaders*:

```
PRESCstReaderCollator_storeSampleData:deserialize sample error in topic 'MyTopic' with type 'MyType'
```

[RTI Issue ID QUEUEING-764]

4.3 [Critical] Queuing Service sent samples with wrong data representation to Consumers

Queuing Service may have forwarded samples with the wrong data representation to Consumers when any of the representations of the *DataReader* in a *SharedReaderQueue* differed from the representation of the *DataWriter* in the same *SharedReaderQueue*.

For example, you may have configured the *DataReader* using the *DataRepresentationQoSPolicy* to accept XCDR and the *DataWriter* to publish XCDR2. When the *DataWriter* published samples with the wrong representation, you may have seen deserialization errors on the application's Consumer receiving the samples. These errors occurred only when the topic type on the application's Consumer limited the number of supported representations using the **allowed_data_representation** annotation. For example:

```
@allowed_data_representation(XCDR2)
struct MyType {
```

4.4 [Major] Lifespan per Producer not applied when Producer sent data on behalf of another Producer's

```
long my_member;  
};
```

If the application *DataReaders* were from a different DDS vendor, you may have seen deserialization errors regardless of the **allowed_data_representation** annotation.

[RTI Issue ID QUEUEING-766]

4.4 [Major] Lifespan per Producer not applied when Producer sent data on behalf of another Producer's DataWriter

Lifespan per Producer was not applied correctly if a Producer sent data on behalf of another *DataWriter*. This could have happened if either of these were true:

- There was an instance of *RTI Routing Service* (with [<publish with original info>](#) for the corresponding *TopicRoute* set to true) acting as a Producer in between the original *DataWriter* and *Queuing Service*.
- A Producer wrote samples on behalf of multiple *DataWriters* using different [Sample Identities](#).

In these scenarios, *Queuing Service* applied the [Lifespan per SharedReader Queue](#) instead, which could have led to incorrect behavior when transitioning samples to the [expired state](#).

The root cause of the problem was that *Queuing Service* internally relied on the **original_publication_virtual_guid** when performing a lookup for Producer. Now the lookup is based on **publication_virtual_guid** instead.

[RTI Issue ID QUEUEING-752]

4.5 [Minor] Warnings when using Queuing Service with replication

When launching *Queuing Service* with replication enabled, you may have seen warnings regarding inputs and outputs that were not enabled.

```
RTI_RoutingServiceRoute_lookup_input_by_name:!input not enabled with name=QueueInput  
RTI_RoutingServiceRoute_lookup_output_by_name:!output not enabled with  
name=RedistributionOutput  
RTI_RoutingServiceRoute_lookup_output_by_name:!output not enabled with  
name=MasterElectionOutput  
RTI_RoutingServiceRoute_lookup_output_by_name:!output not enabled with name=QueueOutput  
RTI_RoutingServiceRoute_lookup_output_by_name:!output not enabled with  
name=RedistributionOutput
```

The documentation for these lookup APIs failed to mention that the inputs/outputs returned must be enabled in order to be retrieved. The API Reference documentation has been updated to explain this, and we have removed these warnings in this release.

This condition (that the inputs/outputs must be enabled) will be removed in a future release (tracked with RTI issue ID ROUTING-1131).

[RTI Issue ID QUEUEING-727]

4.6 [Minor] Fourth digit of product version not logged by Queuing Service at startup

The *Queuing Service* executable did not log the fourth digit (revision) of the product version upon service start. As a result, any patches were indistinguishable from the base version.

[RTI Issue ID QUEUEING-746]

5 Known Issues

5.0.1 Some tags in the XML configuration must be grouped in a strict order

The XML validator tools *Queuing Service* uses to validate XML configuration files adhere to the XML 1.0 specification, which doesn't offer a way of defining collections of unordered tags that are both bounded and unbounded in occurrences.

This limitation is no longer present in XML 1.1. However, there are no C or C++ validators compliant with the XML 1.1 specification at the time of writing.

[RTI Issue ID CORE-14178]

6 Current Limitations

The QueueProducer and QueueConsumer wrapper APIs are only supported for the Modern C++ API.

7 Available Documentation

Queuing Service documentation also includes:

- **Getting Started Guide** (RTI_Queueing_Service_GettingStarted.pdf)—Provides installation and startup instructions.
- **User's Manual** (RTI_Queueing_Service_UsersManual.pdf)—Describes how to configure and use *Queuing Service*.